Operating Systems & Compilers

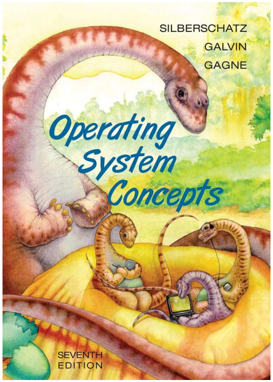
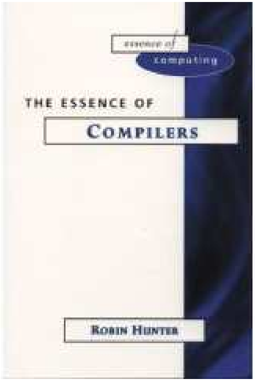
Summing up

Frédéric Haziza <daz@it.uu.se>

Department of Computer Systems
Uppsala University

Spring 2007

Hans Flack


SILBERSCHATZ
GALVIN
GAGNE
Operating System Concepts
SEVENTH EDITION

essence of computing
THE ESSENCE OF
COMPILERS
ROBIN HUNTER

- A. Silberschatz, P. B. Galvin, G. Gagne.
  Operating System Concepts, Seventh Edition.
  Wiley, 2002 (ISBN: 0-471-69466-5)
- R. Hunter.
  The Essence of Compilers.
  Prentice-Hall, 1999 (ISBN: 0-13-727835-7)
- Course Homepage.
  http://www.it.uu.se/edu/course/homepage/oskomp/vt07.

Conficius, 5th century BC

I hear and I forget,
I see and I remember,
I do and I understand.

- You take your own notes
- Those slides are no placeholders for lecture notes

- 2 labs (mandatory)
- 4 handins (bonus)

Providing the environment for programs to run

- To increase CPU utilization: **multitasking**
- 2 process schedulers
- Control is eventually switched back to the system through
  - an interrupt (hardware error detection)
  - a trap (software-generated interrupt) or
  - a system call (interface to ask the OS to perform privileged tasks

Process Management

Resources (CPU time, memory, files, I/O) are either

- given at creation or
- allocated while running.

Definition (Process)

Unit of work in the system. For both user and system.

- Creation / Deletion of processes
- Suspending / Resuming process
- Mechanism for process synchronization
- Mechanism for process communication
- Mechanism for deadlock handling
  (prevention, avoidance, reparation, ...)

⇒ PCB, states, transitions, ready queue, device queue, context switch IPC, message passing, shared Memory, threads, responsiveness, user/kernel-threads, CPU burst cycles, preemptive vs cooperative, throughput, turnaround time, response time, waiting time, FIFO, SJF, Priority Scheduling, RR, Multi-level queues, starvation, aging, time quantum, symmetric/asymmetric processors, processor affinity, localities, semaphores, locks

Deadlock: Resource Allocation Graph, Prevention, Avoidance, Detection, Recovery

Memory Management

- Keeping track of which parts of memory are currently being used and by whom
- Which processes and data to move in and out memory
- Allocating and deallocating memory space as needed

⇒ Address binding, relocation register, linking, dynamic loading, input queue, contiguous memory, xxx-fit algorithms, fragmentation, compaction, paging

Virtual memory: Paging, Segmentation, demand-paging, logical space, physical space, pages, frames, page table, frame table, Page Table Base Register, TLB, shared pages, reentrant code, lazy swapping, page-fault, page replacement algorithms, modify bit, reference string, FIFO, OPT, LRU, LRU with approximations, Belady's anomaly, global vs local page replacement, thrashing, working-set, page-fault frequency

Storage Management

- File system
- Disks
- Protection and Security

$\Rightarrow$ done in the essay

Compiling Process & Lexical analysis

⇒ Analysis stage, synthesis stage, translators, efficiency, correctness, source code, machine code, intermediate representation, lexical analysis, lexer, syntax analysis, parser, semantic analysis, abstract syntax tree, regular expressions, grammars, Finite automaton

Parsing

$\Rightarrow$ tokens, Context-free grammars, LL(k)/LR(k) grammars, productions, derivation, terminals/non-terminals, lookahead symbol, Parse tree, abstract syntax tree, top-down, bottom-up, recursive-descent, shift-reduce, left-factoring, parse table, left-recursion elimination

Semantic analysis & Code Generation

⇒ scope, type, fault-tolerance, symbol table, stack/hashtable/... implementations, name collisions

⇒ three-address code, P-code, Bytecode, JVM, stack, implicit operands, CISC vs RISC, instruction selection, register allocation, liveness, graph-coloring register allocation, optimizations